

Almost Optimal Cover-Free Families

Nader H. Bshouty and Ariel Gabizon

Department of Computer Science
Technion, Haifa, 32000

Abstract. Roughly speaking, an $(n, (r, s))$ -Cover Free Family (CFF) is a small set of n -bit strings such that: “in any $d := r + s$ indices we see all patterns of weight r ”. CFFs have been of interest for a long time both in discrete mathematics as part of block design theory, and in theoretical computer science where they have found a variety of applications, for example, in parametrized algorithms where they were introduced in the recent breakthrough work of Fomin, Lokshtanov and Saurabh [16] under the name ‘lopsided universal sets’.

In this paper we give the first explicit construction of cover-free families of optimal size up to lower order multiplicative terms, *for any r and s* . In fact, our construction time is almost linear in the size of the family. Before our work, such a result existed only for $r = d^{o(1)}$ and $r = \omega(d/(\log \log d \log \log \log d))$. As a sample application, we improve the running times of parameterized algorithms from the recent work of Gabizon, Lokshtanov and Pilipczuk [18].

1 Introduction

The purpose of this paper is to give an explicit almost optimal construction of *cover free families* [20]. Before giving a formal definition, let us describe the special case of *group testing*. The problem of group testing was first presented during World War II and described as follows [10,26]: Among n soldiers, at most s carry a fatal virus. We would like to blood test the soldiers to detect the infected ones. Testing each one separately will give n tests. To minimize the number of tests we can mix the blood of several soldiers and test the mixture. If the test comes negative then none of the tested soldiers are infected. If the test comes out positive, we know that at least one of them is infected. The problem is to come up with a small number of tests.

To obtain a non-adaptive algorithm for this problem, a little thought shows that what is required is a set of tests such that for any subset T of s soldiers, and any soldier $i \notin T$, there is a test including soldier i , and precluding all soldiers in T . Let $d = s + 1$. Viewing a test as a characteristic vector $a \in \{0, 1\}^n$ of the soldiers it includes, the desired property is equivalent to the following. Find a small set $\mathcal{F} \subseteq \{0, 1\}^n$ such that for every $1 \leq i_1 < i_2 < \dots < i_d \leq n$, and every $1 \leq j \leq d$, there is $a \in \mathcal{F}$ such that $a_{i_j} = 1$ and $a_{i_k} = 0$ for all $k \neq j$.

1.1 Cover-Free Families

We can view \mathcal{F} described above as a set of strings such that “in any d indices we see all patterns of weight one”. We can generalize this property by choosing an integer $1 \leq r < d$ and requesting to see “in any d indices all patterns of weight r ”.

Definition 1 (Cover-Free Family). Fix positive integers r, s, n with $r, s < n$ and let $d := r + s$. An $(n, (r, s))$ -Cover Free Family (CFF) is a set $\mathcal{F} \subseteq \{0, 1\}^n$ such that for every $1 \leq i_1 < i_2 < \dots < i_d \leq n$ and every $J \subset [d]$ of size $|J| = r$ there is $a \in \mathcal{F}$ such that $a_{i_j} = 1$ for $j \in J$ and $a_{i_k} = 0$ for $k \notin J$.

We will always assume $r \leq d/2$ (and therefore $r \leq s$): If not, construct an $(n, (s, r))$ -CFF and take the set of complement vectors.

We note that the definition of CFFs usually given is a different equivalent one which we now describe. Given an $(n, (r, s))$ -CFF \mathcal{F} , denote $N = |\mathcal{F}|$ and construct the $N \times n$ boolean matrix A whose rows are the elements of \mathcal{F} . Now, let X be a set of N elements and think of the *columns* of A as characteristic vectors of subsets, which we will call *blocks*, $B \subseteq X$. That is, if we denote by $\mathcal{B} = \{B_1, \dots, B_n\}$ the set of blocks corresponding to these columns, then A is the *incidence matrix* of \mathcal{B} , i.e. the i 'th element of X is in B_j if and only if $A_{i,j} = 1$.

For this view, the CFF property of \mathcal{F} implies the following: For any blocks $B_1, \dots, B_r \in \mathcal{B}$ and any other s blocks $A_1, \dots, A_s \in \mathcal{B}$ (distinct from the B 's), there is an element of X contained in all the B 's but not in any of the A 's, i.e.

$$\bigcap_{i=1}^r B_i \not\subseteq \bigcup_{j=1}^s A_j.$$

This property is the usual way to define CFFs [20].

Notation: Let us denote by $N(n, (r, s))$ the minimal integer N such that there exists an $(n, (r, s))$ -CFF \mathcal{F} of size $|\mathcal{F}| = N$.

1.2 Previous Results

It is known that, [32],

$$N(n, (r, s)) \geq \Omega(N(r, s) \cdot \log n)$$

where

$$N(r, s) := \frac{d \binom{d}{r}}{\log \binom{d}{r}}.$$

Using the union bound it is easy to show that for $d = r + s = o(n)$, $r \leq s$, we have

$$N(n, (r, s)) \leq O\left(\sqrt{r} \log \binom{d}{r} \cdot N(r, s) \cdot \log n\right).$$

D'yachkov et. al.'s breakthrough result, [14], implies that for $s, n \rightarrow \infty$

$$N(n, (r, s)) = \Theta(N(r, s) \cdot \log n). \quad (1)$$

The two above bounds are non-constructive.

It follows from [31], that for an infinite sequence of integers n , an $(n, (r, s))$ -CFF of size

$$M = O\left((rd)^{\log^* n} \log n\right)$$

can be constructed in polynomial time.

Before proceeding to describe previous results and ours, we introduce some convenient terminology:

We will think of the parameter $d = r + s$ as going to infinity and always use the notation $o(1)$ for a term that is independent of n , and goes to 0 as $d \mapsto \infty$.

We say an $(n, (r, s))$ -CFF \mathcal{F} is *almost optimal*, if its size $N = |\mathcal{F}|$ satisfies

$$N = N(r, s)^{1+o(1)} \cdot \log n = \begin{cases} d^{r+1+o(1)} \log n & \text{if } r = O(1) \\ \left(\frac{d}{r}\right)^{r+o(r)} \log n & \text{if } r = \omega(1), r = o(d) \\ 2^{H_2(r/d)d+o(d)} \log n & \text{if } r = O(d) \end{cases}$$

where $H_2(x)$ is the binary entropy function.

We say that such \mathcal{F} can be *constructed in linear time* if it can be constructed in time $O(N(r, s)^{1+o(1)} \cdot \log n \cdot n)$. In this terminology, our goal is to obtain almost optimal CFFs that are constructible in linear time.

Let us first consider the case of constant r . It is not hard to see that in this case an $(n, (r, s))$ -CFF \mathcal{F} of size $d^{r+1} \log n$ is almost optimal by our definition (and in fact exceeds the optimal size in (2) only by a multiplicative $\log d$ factor). Bshouty [8] constructs \mathcal{F} of such size in linear time and thus solves the case of constant r . In fact, calculation shows that for any $r = d^{o(1)}$, \mathcal{F} of size

$$N = 2^{O(r)} \cdot d^{r+1} \cdot \log n$$

is almost optimal. Bshouty [7,8] constructs such \mathcal{F} in linear time for any $r = o(d)$. We proceed to the case of larger r . Fomin et. al. [16] construct an $(n, (r, s))$ -CFF of size

$$\binom{d}{r} 2^{O\left(\frac{d}{\log \log(d)}\right)} \log n \quad (2)$$

in linear time. This is almost optimal when

$$r = \omega\left(\frac{d}{\log \log d \log \log \log d}\right).$$

To the best of our knowledge there is no explicit construction of almost optimal $(n, (r, s))$ -CFFs when $d^{o(1)} < r < \omega(d/(\log \log d \log \log \log d))$.

Note that in this range (and even for $r = \omega(1)$ and $r = o(d)$), \mathcal{F} is almost optimal if and only if it has size

$$N = \binom{d}{r}^{1+o(1)} \log n = \left(\frac{d}{r}\right)^{r(1+o(1))} \cdot \log n.$$

Gabizon et. al [18] made a significant step for general r and constructed an $(n, (r, s))$ -CFF of size

$$O((d/r)^{2 \cdot r} \cdot 2^{O(r)} \cdot \log n)$$

in linear time. This is quadratically larger than optimal.

1.3 New Result

As mentioned before, there is no explicit construction of almost optimal $(n, (r, s))$ -CFFs when $d^{o(1)} < r < \omega(d/(\log \log d \log \log \log d))$ and the result of [18] is quadratically larger than optimal. In this paper we close this quadratic gap and give an explicit construction of an almost optimal $(n, (r, s))$ -CFF for all r and s . Our main result is

Theorem 1. *Fix any integers $r < s < d$ with $d = r + s$. There is an almost optimal $(n, (r, s))$ -CFF, i.e., of size*

$$N(r, s)^{1+o(1)} \cdot \log n,$$

that can be constructed in linear time. That is, in time

$$O(N(r, s)^{1+o(1)} \cdot n \cdot \log n)$$

As we've seen in Section 1.2, the above theorem is already proved for $r < d^{o(1)}$ and $r > \omega(d/(\log \log d \log \log \log d))$.

2 Applications of result

2.1 Application to learning hypergraphs

Let $\mathcal{G}_{s,r}$ be a set of all labeled hypergraphs of rank at most r (the maximum size of an edge $e \subseteq V$ in the hypergraph) on the set of vertices $V = \{1, 2, \dots, n\}$ with at most s edges. Given a hidden Sperner hypergraph¹ $G \in \mathcal{G}_{s,r}$, we need to identify it by asking *edge-detecting queries*. An edge-detecting query $Q_G(S)$, for $S \subseteq V$ is: Does S contain at least one edge of G ? Our objective is to *non-adaptively* learn the hypergraph G by asking as few queries as possible.

This problem has many applications in chemical reactions, molecular biology and genome sequencing, where deterministic non-adaptive algorithms are most desirable. In chemical reactions, we are given a set of chemicals, some of which

¹ The hypergraph is Sperner hypergraph if no edge is a subset of another. If it is not Sperner hypergraph then learning is not possible.

react and some which do not. When multiple chemicals are combined in one test tube, a reaction is detectable if and only if at least one set of the chemicals in the tube reacts. The goal is to identify which sets react using as few experiments as possible. The time needed to compute which experiments to do is a secondary consideration, though it is polynomial for the algorithms we present. See [3] and references within for more details and many other applications in molecular biology.

The above hypergraph $\mathcal{G}_{s,r}$ learning problem is equivalent to the problem of exact learning a monotone DNF with at most s monomials (monotone terms), where each monomial contains at most r variables (s -term r -MDNF) from membership queries [1,4]. A membership query, for an assignment $a \in \{0,1\}^n$ returns $f(a)$ where f is the hidden s -term r -MDNF.

The non-adaptive learnability of s -term r -MDNF was studied in [33,24,25,17,11,9]. All the algorithms are either deterministic algorithms that uses non-optimal constructions of $(n, (s, r))$ -CFF or randomized algorithms that uses randomized constructions of $(n, (s, r))$ -CFF. Our construction in this paper gives, for the deterministic algorithm, a better query complexity and changes the randomized algorithm to deterministic. Recently, our construction is used in [3] to give a polynomial time almost optimal algorithm for learning $\mathcal{G}_{s,r}$.

2.2 Application to r -Simple k -Path

Gabizon et. al. [18] recently constructed deterministic algorithms for parametrized problems with ‘relaxed disjointness constraints’. For example, rather than searching for a *simple* path of length k in a graph of n vertices, we can search for a path of length k where no vertex is visited more than r times, for some ‘relaxation parameter’ r . We call the problem of deciding whether such a path exists r -SIMPLE k -PATH. Abasi et. al [2] were the first to study r -SIMPLE k -PATH and presented a randomized algorithm running in time $O^*(r^{2k/r})$. What is perhaps surprising, is that the running time can *significantly improve* as r grows. Derandomizing the result of [2], [18] obtained a deterministic algorithm for r -SIMPLE k -PATH with running time $O^*(r^{12k/r} \cdot 2^{O(k/r)})$. At the core of their derandomization is the notion of a ‘multiset separator’ - a small family of ‘witnesses’ for the fact that two multisets do not ‘intersect too much’ on any particular element. How small this family of witnesses can be in turn depends on how small an $(n, (2k/r, k - 2k/r))$ -CFF one can construct (details on these connections are given in Appendix B). Plugging in our new construction into the machinery of [18], we get

Theorem 2. r -SIMPLE k -PATH can be solved in deterministic time $O(r^{8k/r+o(k/r)} \cdot 2^{O(k/r)} \cdot k^{O(1)} \cdot n^3 \cdot \log n)$.

For example, when both k/r and r tend to infinity, we get running time $O^*(r^{8k/r+o(k/r)})$ and [18] get $O^*(r^{12k/r+o(k/r)})$.

In a well-known work, Koutis [21] observed that practically all parametrized problems can be viewed as special cases of ‘multilinear monomial detection’. [18] also studied the relaxed version of this more general problem: Given an arithmetic circuit C computing an n -variate polynomial $f \in \mathbb{Z}[X_1, \dots, X_n]$, determine

whether f contains a monomial of total degree k and individual degree at most r . We call this problem (r, k) -MONOMIAL DETECTION. [18] define such a circuit C to be *non-canceling* if it contains only variables at its leaves (i.e., no constants), and only addition and multiplication gates (i.e., no subtractions). [18] showed that for non-canceling C , (r, k) -MONOMIAL DETECTION can be solved in time $O^*(|C| \cdot r^{18k/r} \cdot 2^{O(k/r)})$. We obtain

Theorem 3. *Given a non-canceling arithmetic circuit C computing $f \in \mathbb{Z}[X_1, \dots, X_n]$, (r, k) -MONOMIAL DETECTION can be solved in deterministic time $O(|C| \cdot r^{12k/r + o(k/r)} \cdot 2^{O(k/r)} \cdot k^{O(1)} \cdot n^3 \cdot \log n)$.*

Organization of paper

In Section 3 we give an informal description of our CFF construction. In Section 4 we give a simple construction that proves Theorem 1 for any $\log^2 d \leq r \leq d/(\log \log d)^{\omega(1)}$. In Section 5, we give the proof for $d/(\log d)^{\omega(1)} \leq r \leq d/\omega(1)$. The proofs of Theorems 2 and 3 appear in Appendix B.

3 Proof Overview

Our construction is essentially a generalization of [18] allowing a more flexible choices of parameters. For simplicity, we first describe the construction of [18] and then explain our improvements.

To illustrate the ideas in a simple way, the following ‘adaptive’ viewpoint will be convenient: We are given two disjoint subsets $C, D \subseteq [n]$ of sizes $|C| = r$ and $|D| = s$. We wish to divide $[n]$ into two separate buckets such that all elements of C fall into the first, and all elements of D fall into the second. Of course the point in CFFs is *that we do not know C and D in advance*. However, the number of different possibilities for the division that will come up in the process will be a bound on the size of an analogous $(n, (r, s))$ -CFF- which will contain a vector $a \in \{0, 1\}^n$ corresponding to each way of separating $[n]$ into two buckets that came up in the adaptive process.

As a first step we use a perfect hash function h to divide $[n]$ into r buckets such that each bucket contains exactly one element of C . Using a construction of Naor et. al [28], h can be chosen from a family of size $2^{O(r)} \cdot \log n$. Let us call these buckets B_1, \dots, B_r . Now, suppose that we knew, for each $i \in [r]$, the number of elements s_i from D that fell into bucket B_i . In that case we could use an $(n, (1, s_i))$ -CFF \mathcal{F}_i to separate the element of C in B_i from the s_i elements of D , and put each in the correct final bucket.

We have such \mathcal{F}_i of size $c \cdot s_i^2 \cdot \log n$ for universal constant c . Thus, the number of different choices in all buckets is

$$\prod_{i=1}^r c \cdot s_i^2 \cdot \log n \leq c^r \cdot (s/r)^{2r} \cdot \log^r n,$$

as the product of the s_i ’s is maximized when $s_1 = \dots s_r = s/r$. Furthermore, [18] show this can be improved to roughly $(s/r)^r \cdot \log n \leq (d/r)^r \cdot \log n$ where

$d = r + s$. This is done using the hitting sets for combinatorial rectangles of Linial et. al [22] (we do not go into details on this stage here). Of course, we do not know the s_i 's. However, it is not too costly to simply guess them! Or rather, try all options: The number of choices for non-negative integers s_1, \dots, s_r such that $s_1 + \dots + s_r = s$ is at most

$$\binom{d-1}{r-1} \leq \binom{d}{r} \leq (ed/r)^r.$$

Combining all stages, this gives us an $(n, (r, s))$ -CFF of size roughly $(d/r)^{2r+O(1)}$. log n . To get an almost optimal construction, we need to get the 2 in the exponent down to a 1. We achieve this by reducing the cost of the ‘guessing stage’. Instead of r buckets, we begin by dividing $[n]$ into k buckets for some $k = o(r)$, such that every bucket will contain r/k elements of C . This is done using *splitters* [28]. For concreteness, think of $k = r/\log \log d$. (In the final construction we need to choose k more delicately). Now as we only have k s_i 's, there will be less possibilities to go over such that $s_1 + \dots + s_k = s$ - specifically less than $(ed/k)^k$. On the other hand, our task in each bucket is now more costly - we need to separate r/k elements of C from s_i elements of D , rather than just *one* element of C . A careful choice of parameters show this process can be done while going over at most $(d/r)^{1+o(1)}$ options for the partition into two buckets. There are now two main technical issues left to deal with.

- The splitter construction of [28] was not analyzed as being almost-linear time, but rather, only polynomial time. We give a more careful analysis of it's runtime.
- We need to generalize a component from the construction of [18], into what we call “multi-CFFs”. Roughly speaking, this is a small set of strings of length $n \cdot \ell$ that are ‘simultaneously a CFF on each n -bit block’. That is, if we think of the string as divided into ℓ blocks of length n , and wish to see in each block a certain pattern of weight r_i in some subset of d_i indices of that block, there will be one string in the multi-CFF that simultaneously exhibits all patterns. We construct a small multi-CFF using a combination of “dense separating hash functions” and the hitting sets for combinatorial rectangles of [22]. See Section 5 for details.

4 The First Construction

In this section we give the first construction

4.1 Preliminary Results for the First Construction

We begin by giving some definitions and preliminary results that we will need for our first construction. The results in this subsection are from [28] and [8].

Let n, q and d be integers. Let \mathcal{F} be a set of boolean functions $f : [q]^d \rightarrow \{0, 1\}$. Let H be a family of functions $h : [n] \rightarrow [q]$. We say that H is an

(n, \mathcal{F}) -restriction family $((n, \mathcal{F})$ -RF) if for every $\{i_1, \dots, i_d\} \subseteq [n]$, $1 \leq i_1 < i_2 < \dots < i_d \leq n$ and every $f \in \mathcal{F}$ there is a function $h \in H$ such that $f(h(i_1), \dots, h(i_d)) = 1$.

We say that a construction of an (n, \mathcal{F}) -restriction family H is a *linear time construction*, if it runs in time $\tilde{O}(|H| \cdot n) = |H| \cdot n \cdot \text{poly}(\log |H|, \log n)$.

Let H be a family of functions $h : [n] \rightarrow [q]$. For $d \leq q$ we say that H is an (n, q, d) -perfect hash family $((n, q, d)$ -PHF) if for every subset $S \subseteq [n]$ of size $|S| = d$ there is a hash function $h \in H$ such that $h|_S$ is injective (one-to-one) on S , i.e., $|h(S)| = d$. Obviously, an (n, q, d) -PHF is an (n, \mathcal{F}) -RF when $\mathcal{F} = \{f\}$, for some $f : [q]^d \rightarrow \{0, 1\}$ satisfying $f(\sigma_1, \dots, \sigma_d) = 1$ iff $\sigma_1, \dots, \sigma_d$ are distinct.

In [8] Bshouty proved

Lemma 1. *Let q be a power of prime. If $q > 4(d(d-1)/2 + 1)$ then there is a linear time construction of an (n, q, d) -PHF of size*

$$O\left(\frac{d^2 \log n}{\log(q/d^2)}\right).$$

The following is a folklore result

Lemma 2. *Let \mathcal{F} be a set of boolean functions $f : [q]^d \rightarrow \{0, 1\}$. If there is a linear time construction of an (m, \mathcal{F}) -RF where $m > 4(d(d-1)/2 + 1)$ of size s then there is a linear time construction of an (n, \mathcal{F}) -RF of size*

$$O\left(\frac{sd^2 \log n}{\log(m/d^2)}\right).$$

Proof. Let H_1 be an (m, \mathcal{F}) -RF and let H_2 be the (n, m, d) -PHF constructed in Lemma 1. Then it is easy to see that $H_1(H_2) := \{h_1(h_2) \mid h_2 \in H_2, h_1 \in H_1\}$ is an (n, \mathcal{F}) -RF. \square

Another restriction family that will be used here is splitters [28]. An (n, r, k) -splitter is a family of functions H from $[n]$ to $[k]$ such that for all $S \subseteq [n]$ with $|S| = r$, there is $h \in H$ that splits S perfectly, i.e., for all $j \in [k]$, $|h^{-1}(j) \cap S| \in \{\lfloor r/k \rfloor, \lceil r/k \rceil\}$. Obviously, an (n, q, d) -PHF is an (n, d, q) -splitter. Define

$$\sigma(r, k) := \left(\frac{2\pi r}{k}\right)^{k/2} e^{k^2/(12r)}. \quad (3)$$

From the union bound it can be shown that there exists an (n, r, k) -splitter of size $O(\sqrt{r}\sigma(r, k) \log n)$, [28]. Naor et. al, [28], use the r -wise independent probability space to construct an (m, r, k) -splitter. They show

Lemma 3. *For $k \leq r$, an (m, r, k) -splitter of size $O(\sqrt{r}\sigma(r, k) \log m)$ can be constructed in time*

$$O(\sqrt{r} \cdot \sigma(r, k) m^{2r} \log m).$$

When $k = \omega(\sqrt{r})$, Naor et. al. in [28], constructed an (n, r, k) -splitter of size $O(\sigma(r, k)^{1+o(1)} \log n)$ in polynomial time. We here show that the same construction can be done in *linear time*. They first construct an $((r/z)^2, r/z, k/z)$ -splitter using Lemma 3 where $z = \Theta(r \log k / (k \log(2r/k)))$. They then use Lemma 2 to construct an $(r^2, r/z, k/z)$ -splitter. Then compose z pieces of the latter to construct an (r^2, r, k) -splitter and then again use Lemma 2 to construct the final (n, r, k) -splitter.

Note here that we assume that $z|k|r$. The result can be extended to any z, k and r .

We now prove

Lemma 4. *For $k = \omega(\sqrt{r})$ and $z = 16r \log k / (k \log(4r/k))$. An (n, r, k) -splitter of size*

$$r^{O(z)} \sigma(r, k) \log n = \sigma(r, k)^{1+o(1)} \log n$$

can be constructed in time $O(\sigma(r, k)^{1+o(1)} \log n)$.

Proof. By Lemma 11 in Appendix A, z is a monotonic decreasing function in k and $16\sqrt{r} \geq z \geq 8 \log r$ for $\sqrt{r} \leq k \leq r$. First we construct an $((r/z)^2, r/z, k/z)$ -splitter using Lemma 3. By Lemma 3 and Lemma 12 in Appendix A, this takes time

$$O(\sqrt{r/z} \cdot \sigma(r/z, k/z) ((r/z)^2)^{2r/z} \log(r/z)) = o(\sigma(r, k)).$$

By Lemma 3, the size of this splitter is $O(\sqrt{r/z} \cdot \sigma(r/z, k/z) \log(r/z))$. By Lemma 2, using the above splitter, an $(r^2, r/z, k/z)$ -splitter H of size

$$O((r/z)^{2.5} \sigma(r/z, k/z) \log(r/z) \log r)$$

can be constructed in linear time. Now, for every choice of $0 = i_0 < i_1 < i_2 < \dots < i_{z-1} < i_z = r^2$ and $h_0, h_1, \dots, h_{z-1} \in H$ define the function $h(j) = h_t(j) + (k/z)t$ if $i_t < j \leq i_{t+1}$. It is easy to see that this gives an (r^2, r, k) -splitter. The splitter can be constructed in linear time and by Lemma 13 in Appendix A, its size is

$$\binom{r^2}{z-1} (c_1 (r/z)^{2.5} \sigma(r/z, k/z) \log(r/z) \log r)^z = r^{c_2 z} \sigma(r, k)$$

for some constants c_1 and c_2 . Now by Lemma 2 and Lemma 14 in Appendix A, an (n, r, k) -splitter can be constructed in time

$$O(r^2 (r^{c_2 z} \sigma(r, k)) \log n) = r^{O(z)} \sigma(r, k) \log n = \sigma(r, k)^{1+o(1)} \log n.$$

□

The following is from [8]

Lemma 5. *There is an $(n, (r, s))$ -CFF of size*

$$O\left(rs \binom{2rs}{r} \log n\right)$$

that can be constructed in linear time.

4.2 Construction I

Let $r \leq s$ be integers and $d = r + s$. Obviously, $1 \leq r \leq d/2$ and $d/2 \leq s \leq d$. We may also assume that

$$r > \text{poly}(\log d) = d^{o(1)}. \quad (4)$$

See the table in Section 1.2 and the discussion following it.

We first use Lemma 2 to reduce the problem to constructing a $(q, (r, s))$ -CFF for $q = O(d^3)$. We then do the following. Suppose $1 \leq i_1 < i_2 < \dots < i_d \leq q$ and let $(\xi_1, \dots, \xi_d) \in \{0, 1\}^d$ with r ones (and s zeros) that is supposed to be assigned to (i_1, i_2, \dots, i_d) . Let i_{j_1}, \dots, i_{j_r} be the entries for which $\xi_{j_1}, \dots, \xi_{j_r}$ are equal to 1. The main idea of the construction is to first deal with entries i_{j_1}, \dots, i_{j_r} that are assigned to one and distribute them equally into k buckets, where k will be determined later. This can be done using a (q, r, k) -splitter. Each bucket will contain r/k ones and an unknown number of zeros. We do not know how many zeros, say $d_i - (r/k)$, fall in bucket i but we know that $d_1 + \dots + d_k = d$. That is, bucket i contains d_i indices of i_1, i_2, \dots, i_d for which r/k of them are ones. We take all possible $d_1 + \dots + d_k = d$ and for each bucket i construct $(q, d_i - (r/k), r/k)$ -CFF. Taking all possible functions in each bucket for each possible $d_1 + \dots + d_k = d$ solves the problem.

Let H_1 be an (n, q, d) -PHF such that $d^3 < q \leq 2d^3$ is a power of prime and $d = r + s$. The following follows from Lemma 2

Lemma 6. *If H is a $(q, (r, s))$ -CFF then $\{h_1(h) \mid h_1 \in H_1, h \in H\}$ is $(n, (r, s))$ -CFF of size $|H| \cdot |H_1|$.*

We now construct a $(q, (r, s))$ -CFF. Let H_2 be a (q, r, k) -splitter where $k < r$ will be determined later. Let $H'_3[d']$ and $H''_3[d']$ be a $(q, d' - \lfloor r/k \rfloor, \lfloor r/k \rfloor)$ -CFF and $(q, d' - \lceil r/k \rceil, \lceil r/k \rceil)$ -CFF respectively and define $H_3[d'] := H'_3[d'] \cup H''_3[d']$ where $d \geq d' \geq \lceil r/k \rceil$. For every $(h_1, \dots, h_k) \in H_3[d_1] \times \dots \times H_3[d_k]$ where $d_1 + \dots + d_k = d$ and $g \in H_2$ define the function

$$H_{h_1, \dots, h_k, g}(i) = h_{g(i)}(i).$$

We first prove

Lemma 7. *The set of all $H_{h_1, \dots, h_k, g}$ where $(h_1, \dots, h_k) \in H_3[d_1] \times \dots \times H_3[d_k]$ for some $d_1 + \dots + d_k = d$ and $g \in H_2$ is a $(q, (r, s))$ -CFF.*

Proof. Consider any $1 \leq i_1 < i_2 < \dots < i_d \leq q$ and any (ξ_1, \dots, ξ_d) of weight r . Let $S = \{i_1, \dots, i_d\}$. Consider $I = \{i_j \mid \xi_j = 1\}$. Since H_2 is a (q, r, k) -splitter there is $g \in H_2$ such that $|g^{-1}(j) \cap I| \in \{\lfloor r/k \rfloor, \lceil r/k \rceil\}$ for all $j = 1, \dots, k$. Let $d_j = |g^{-1}(j) \cap S|$ for $j = 1, \dots, k$. Then $d_1 + d_2 + \dots + d_k = d$. Since $H_3[d_j]$ is a $(q, d_j - \lfloor r/k \rfloor, \lfloor r/k \rfloor)$ -CFF and $(q, d_j - \lceil r/k \rceil, \lceil r/k \rceil)$ -CFF, there is $h_j \in H_3[d_j]$ such that $h_j(g^{-1}(j) \cap I) = \{1\}$ and $h_j(g^{-1}(j) \cap (S \setminus I)) = \{0\}$.

Now, if $\xi_\ell = 1$ then $i_\ell \in I$. Suppose $g(i_\ell) = j$. Then $i_\ell \in g^{-1}(j) \cap I$ and

$$H_{h_1, \dots, h_k, g}(i_\ell) = h_j(i_\ell) \in h_j(g^{-1}(j) \cap I) = \{1\}.$$

If $\xi_\ell = 0$ then $i_\ell \in S \setminus I$. Suppose $g(i_\ell) = j$. Then $i_\ell \in g^{-1}(j) \cap (S \setminus I)$ and

$$H_{h_1, \dots, h_k, g}(i_\ell) = h_j(i_\ell) \in h_j(g^{-1}(j) \cap (S \setminus I)) = \{0\}.$$

4.3 Size of Construction I

We now analyze the size of the construction. We will use c_1, c_2, \dots for constants that are independent of r, s and n .

Let $d^3 < q \leq 2d^3$ be a power of prime. By Lemma 6 and Lemma 7 the size of the construction is

$$N := |H_1| \cdot |H_2| \cdot \left| \bigcup_{d_1 + \dots + d_k = d} H_3[d_1] \times \dots \times H_3[d_k] \right|$$

where H_1 is an (n, q, d) -PHF, H_2 is a (q, r, k) -splitter and $H_3[d']$ is a $(q, d' - \lceil r/k \rceil, \lceil r/k \rceil)$ -CFF and $(q, d' - \lfloor r/k \rfloor, \lfloor r/k \rfloor)$ -CFF.

Let $z = 16r \log k / (k \log(4r/k))$. By Lemma 4,1 and 5 we have

$$\begin{aligned} N &\leq c_1 \frac{d^2 \log n}{\log d} \cdot r^{O(z)} \sigma(r, k) (\log d) \cdot \\ &\quad \sum_{d_1 + \dots + d_k = d} \prod_{i=1}^k c_2 \frac{d_i r}{k} \binom{2d_i \lceil r/k \rceil}{\lceil r/k \rceil} \log d \\ &\leq c_1 d^2 r^{O(z)} \left(\frac{2\pi r}{k} \right)^{k/2} e^{k^2/(12r)} (\log n) \cdot \\ &\quad c_3^k \left(\frac{r \log d}{k} \right)^k \sum_{d_1 + \dots + d_k = d} \prod_{i=1}^k (2ed_i)^{r/k+1} d_i \tag{5} \\ &\leq c_4^k d^2 r^{O(z)} e^{k^2/(12r)} \left(\frac{r^3 \log^2 d}{k^3} \right)^{k/2} (2e)^r (\log n) \sum_{d_1 + \dots + d_k = d} \prod_{i=1}^k d_i^{r/k+2} \\ &\leq c_5^k d^2 r^{O(z)} e^{k^2/(12r)} \left(\frac{r^3 \log^2 d}{k^3} \right)^{k/2} (2e)^r (\log n) \left(\frac{d}{k} \right)^k \max_{d_1 + \dots + d_k = d} \left(\prod_{i=1}^k d_i \right)^{r/k+2} \tag{6} \\ &\leq c_6^k d^2 r^{O(z)} e^{k^2/(12r)} \left(\frac{r^3 \log^2 d}{k^3} \right)^{k/2} (2e)^r \left(\frac{d}{k} \right)^{r+3k} \log n \tag{7} \\ &\leq c_6^k d^2 r^{O(z)} e^{k^2/(12r)} \left(\frac{r^3 d^6 \log^2 d}{k^9} \right)^{k/2} \left(\frac{2er}{k} \right)^r \left(\frac{d}{r} \right)^r \log n \end{aligned}$$

(5) follows from (3) and the fact that $\binom{a}{b} \leq (ea/b)^b$. (6) follows from the fact that the number of k -tuples (d_1, \dots, d_k) such that $d_1 + \dots + d_k = d$ is $\binom{d+k-1}{k-1} \leq c^k (d/k)^k$ for some constant c . (7) follows from the fact that $\max_{d_1 + \dots + d_k = d} \prod_{i=1}^k d_i = (d/k)^k$.

In summary, we have

$$N \leq c_6^k d^2 r^{O(z)} e^{k^2/(12r)} \left(\frac{r^3 d^6 \log^2 d}{k^9} \right)^{k/2} \left(\frac{2er}{k} \right)^r \left(\frac{d}{r} \right)^r \log n.$$

Now assume $r > \log^2 d$ (see (4)) and let $k := r / \log \log d$.

Since

$$z \log r = \frac{16r \log k \log r}{k \log(4r/k)} \leq c_7 \frac{\log^2 r \log \log d}{\log \log \log d} = o(r),$$

$$\frac{k^2}{12r} = \frac{r}{12(\log \log d)^2} = o(r)$$

$$\left(\frac{r^3 d^6 \log^2 d}{k^9} \right)^{k/2} = c_8^r \left(\frac{d}{r} \right)^{3k} = c_8^r \left(\frac{d}{r} \right)^{o(r)},$$

and $d/r \geq 2$, we have,

$$N \leq (c_9 \log \log d)^r \left(\frac{d}{r} \right)^{r(1+o(1))} \log n.$$

This is

$$\left(\frac{d}{r} \right)^{r(1+o(1))} \log n = N(r, s)^{1+o(1)} \log n$$

when

$$\log^2 d \leq r \leq \frac{d}{(\log \log d)^{\omega(1)}}.$$

5 The Second Construction

In the second construction we replace each component $H_3[d_1] \times \dots \times H_3[d_k]$ with another construction that is built from scratch and therefore has smaller size. The main idea is the following: rather than taking all possible functions in each $(q, d_i - (r/k), r/k)$ -CFF in each bucket, we construct what we call a “multi-CFF”. We first construct a dense “separating hash family” that maps the entries to a smaller domain $[q]$ and separates entries that are supposed to be assigned zero from those that are supposed to be assigned one (i.e., they are mapped to disjoint sets). This is done in each bucket. We then use the hitting set for dense combinatorial rectangles of Linial et. al, [22], to give a separating hash family for all the buckets. Then we build a multi-CFF by assigning 0 and 1 to every possible two disjoint sets. We proceed with the details of the second construction.

5.1 Preliminary Results For the Second Construction

Let H be a set of functions $h : [n] \rightarrow [q]$. We say that H is a $(1 - \epsilon)$ -dense $(n, q, (\rho_1, \rho_2))$ -Separating Hash Family (SHF) if for every two disjoint subsets $S_1, S_2 \subseteq [n]$ of sizes $|S_1| = \rho_1, |S_2| = \rho_2$ there are at least $(1 - \epsilon)|H|$ hash functions $h \in H$ such that $h(S_1) \cap h(S_2) = \emptyset$.

The following lemma follows from [8].

Lemma 8. *Let q be a power of prime. If $\epsilon > 4(\rho_1\rho_2 + 1)/q$ then there is a $(1 - \epsilon)$ -dense $(n, q, (\rho_1, \rho_2))$ -SHF of size*

$$O\left(\frac{\rho_1\rho_2 \log n}{\epsilon \log(\epsilon q / e(\rho_1\rho_2 + 1))}\right)$$

that can be constructed in linear time.

Let $R \subseteq [t]^k$ be a set of the form $R_1 \times \dots \times R_k$, where $R_i \subseteq [t]$. We say R is a *combinatorial rectangle with sidewise density γ* , if for every $i \in [k]$, $|R_i| \geq \gamma \cdot t$. A set $H \subseteq [t]^k$ is called a *hitting set for rectangles with sidewise density γ* if for every set $R \subseteq [t]^k$ that is a combinatorial rectangle of sidewise density γ , $R \cap H \neq \emptyset$.

Linial et. al [22] gave the following construction of a hitting set for combinatorial rectangles.

Lemma 9. *A hitting set for rectangles $H \subseteq [t]^k$ with sidewise density $1/3$ of size $|H| = t^{O(1)} \cdot 2^{O(k)}$ can be constructed in time $t^{O(1)} \cdot 2^{O(k)}$.*

Let H be a set of functions $h : [k] \times [n] \rightarrow \{0, 1\}$. We say that H is an $(n, ((\rho_{1,1}, \rho_{1,2}), \dots, (\rho_{k,1}, \rho_{k,2})))$ -Multi-CFF (MCFF) if for every k pairs of disjoint subsets $(S_{i,1}, S_{i,2}) \subseteq [n]$ of sizes $|S_{i,1}| = \rho_{i,1}, |S_{i,2}| = \rho_{i,2}$, $i = 1, \dots, k$, there is $h \in H$ such that $h(i, S_{i,1}) = 1$ and $h(i, S_{i,2}) = 0$ for all $i = 1, \dots, k$.

We now prove

Lemma 10. *There is an $(n, ((\rho_{1,1}, \rho_{1,2}), \dots, (\rho_{k,1}, \rho_{k,2})))$ -MCFF of size*

$$(2^k (\log n) \max_i \rho_{i,1} \rho_{i,2})^{O(1)} \prod_{i=1}^k \binom{48\rho_{i,1}\rho_{i,2}}{\rho_{i,1}}$$

that can be constructed in time $n \times \text{poly}((\max_i \rho_{i,1}\rho_{i,2})2^k \log n)$

Proof. We first choose integers $q_i, i = 1, \dots, k$ that are powers of primes $24\rho_{i,1}\rho_{i,2} < q_i \leq 48\rho_{i,1}\rho_{i,2}$. Since $4(\rho_{i,1}\rho_{i,2})/q_i < 1/2$, by Lemma 8, there is a $1/2$ -dense $(n, q_i, (\rho_{i,1}\rho_{i,2}))$ -SHF H_i of size $|H_i| = t = O((\max_i \rho_{i,1}\rho_{i,2})(\log n))$. Let $H_i = \{h_{i,1}, \dots, h_{i,t}\}$. Let $G \subseteq [t]^k$ be a hitting set for rectangles with sidewise density $1/3$ of size $|G| = t^{O(1)} \cdot 2^{O(k)}$. By Lemma 9 this set can be constructed in time $t^{O(1)} \cdot 2^{O(k)} = \text{poly}((\max_i \rho_{i,1}\rho_{i,2})2^k \log n)$.

Now for every $g \in G$ and every $R_i \subset [q_i]$, of size $|R_i| = \rho_{i,1}$, $i = 1, \dots, k$, consider the functions $h_{1,g_1}, h_{2,g_2}, \dots, h_{t,g_t}$ and define $h : [k] \times [n] \rightarrow \{0, 1\}$ as follows: $h(i, j) = 1$ iff $h_{i,g_i}(j) \in R_i$.

To show that the set of all such h is an $(n, ((\rho_{1,1}, \rho_{1,2}), \dots, (\rho_{k,1}, \rho_{k,2})))$ -MCFF, consider k pairs of disjoint subsets $(S_{i,1}, S_{i,2}) \subseteq [n]$ of sizes $|S_{i,1}| = \rho_{i,1}, |S_{i,2}| = \rho_{i,2}$, $i = 1, \dots, k$. Let $H_i^* = \{h' \in H_i \mid h'(S_{i,1}) \cap h'(S_{i,2}) = \emptyset\}$. Since H_i is a $1/2$ -dense $(n, q_i, (\rho_{i,1}\rho_{i,2}))$ -SHF, we have $|H_i^*| \geq |H_i|/2$. Since $G \subseteq [t]^k$ is a hitting set for rectangles with sidewise density $1/3$ there is $g \in G$ such that $h_{i,g_i} \in H_i^*$ for all $i = 1, \dots, k$. Let R_i be any set of size $\rho_{i,1}$ such that $h_{i,g_i}(S_{i,1}) \subseteq R_i \subseteq [q_i] \setminus h_{i,g_i}(S_{i,2})$. Then the function h defined above satisfies the following: since

$h_{i,g_i}(S_{i,1}) \subseteq R_i$ we have $h(i, S_{i,1}) = 1$ and since $R_i \cap h_{i,g_i}(S_{i,2}) = \emptyset$ we have $h(i, S_{i,2}) = 0$ for all $i = 1, \dots, k$.

The number of such functions h is

$$|G| \prod_{i=1}^k \binom{q_i}{|R_i|}.$$

5.2 Analysis for Construction II

In the analysis we just replace the size of $H_3[d_1] \times \dots \times H_3[d_k]$ in the analysis of construction I to the new size of a $(q, ((d_1 - r/k, r/k), \dots, (d_k - r/k, r/k)))$ -MCFF in Lemma 10 where $d^3 < q \leq 2d^3$ and get

$$\begin{aligned} N &\leq c_1 \frac{d^2 \log n}{\log d} \cdot r^{O(z)} \sigma(r, k) (\log d) \cdot \\ &\quad \sum_{d_1 + \dots + d_k = d} 2^{O(k)} (\log d)^{O(1)} \left(\frac{dr}{k} \right)^{O(1)} \prod_{i=1}^k \binom{c_2 d_i \lceil r/k \rceil}{\lceil r/k \rceil} \\ &\leq c_3^k d^{O(1)} r^{O(z)} \left(\frac{2\pi r}{k} \right)^{k/2} e^{k^2/(12r)} \log n \sum_{d_1 + \dots + d_k = d} \prod_{i=1}^k (c_4 d_i)^{r/k+1} \\ &\leq c_4^k d^{O(1)} r^{O(z)} e^{k^2/(12r)} \left(\frac{r}{k} \right)^{k/2} c_5^r \log n \sum_{d_1 + \dots + d_k = d} \prod_{i=1}^k d_i^{r/k+1} \\ &\leq c_6^k d^{O(1)} r^{O(z)} e^{k^2/(12r)} \left(\frac{r}{k} \right)^{k/2} c_5^r \log n \left(\frac{d}{k} \right)^k \max_{d_1 + \dots + d_k = d} \left(\prod_{i=1}^k d_i \right)^{r/k+1} \\ &\leq c_6^k d^{O(1)} r^{O(z)} e^{k^2/(12r)} \left(\frac{r}{k} \right)^{k/2} c_5^r \left(\frac{d}{k} \right)^{r+2k} \log n \\ &\leq c_6^k d^{O(1)} r^{O(z)} e^{k^2/(12r)} \left(\frac{r}{k} \right)^{k/2} c_5^r \left(\frac{r}{k} \right)^{r+2k} \left(\frac{d}{r} \right)^{r+2k} \log n \end{aligned}$$

Now let $r > \log^2 d$ and $k = r/\varphi(d)$ where $\varphi(d) < \log d$ and $\varphi(d) = \omega(1)$. Then $k = o(r)$

$$c_4^k d^{O(1)} e^{k^2/(12r)} \left(\frac{r}{k} \right)^{k/2} = 2^{O(\frac{r \log \varphi(d)}{\varphi(d)})} = 2^{o(r)}$$

and

$$r^{O(z)} = r^{O((r/k) \log k / \log(2r/k))} = 2^{O(\varphi(d) \log d / \log \varphi(d))} = 2^{o(r)}.$$

Therefore

$$N = (c_7 \varphi(d))^{r+o(r)} \left(\frac{d}{r} \right)^{r+o(r)} \log n$$

which is

$$N(r, s)^{r(1+o(1))} \log n$$

when

$$r = \frac{d}{\varphi(d)^{\omega(1)}}.$$

Since $\varphi(d) < \log d$ is any function that satisfies $\omega(1)$, the above is true for any

$$\frac{d}{(\log d)^{\omega(1)}} \leq r \leq \frac{d}{\omega(1)}.$$

References

1. D. Angluin. Queries and Concept Learning. *Machine Learning*. 2(4), pp. 319–342, (1987).
2. H. Abasi, N. H. Bshouty, A. Gabizon, and E. Haramaty. On r -Simple k -Path. In *MFCS 2014 (Part II)*, pages 1–12, 2014.
3. H. Abasi, N. H. Bshouty, H. Mazzawi. Non-Adaptive Learning of a Hidden Hypergraph ALT 2015 and *CoRR*, abs/arXiv:1502.04137, 2015.
4. D. Angluin, J. Chen. Learning a Hidden Graph using $O(\log n)$ Queries per Edge. *J. Comput. Syst. Sci.* 74(4). pp. 546–556. (2008).
5. N. Alon, R. Yuster, and U. Zwick. Color coding. In *Encyclopedia of Algorithms*. 2008.
6. D. Boneh, J. Shaw. Collusion-Secure Fingerprinting for Digital Data. *IEEE Transactions on Information Theory*, 44(5), pp. 1897–1905, (1998).
7. N. H. Bshouty. Testers and their applications. ITCS 2014, pp. 327–352. (2014). Full version: Electronic Colloquium on Computational Complexity (ECCC) 19: 11. (2012).
8. N. H. Bshouty. Linear time Constructions of some d -Restriction Problems. CIAC 2015. pp. 74–88.
9. F. Y. L. Chin, H. C. M. Leung, S.-M. Yiu. Non-adaptive complex group testing with multiple positive sets. *Theor. Comput. Sci.* 505. pp. 11–18. (2013).
10. D. Z. Du, F. K. Hwang. Combinatorial group testing and its applications. Volume 12 of Series on Applied Mathematics. World Scientific, New York, second edition, (2000).
11. D. Z. Du, F. Hwang. Pooling Design and Nonadaptive Group Testing: Important Tools for DNA Sequencing. World Scientific, Singapore (2006).
12. A. G. D’yachkov and V. V. Rykov. Bounds on the length of disjunctive codes. *Problemy Peredachi Inf*, 18(3), pp. 7–13. (1982).
13. A. G. D’yachkov, V. V. Rykov, A. M. Rashad. Superimposed distance codes. Problems Control Inform. Theory/Problemy Upravlen. Teor. Inform., 18(4), pp. 237–250. (1989).
14. A. G. D’yachkov, I. V. Vorobev, N. A. Polyansky, V. Yu. Shchukin. Bounds on the rate of disjunctive codes. Problems of Information Transmission. 50(1), pp. 27–56. (2014).
15. Z. Füredi. On r -cover-free families. *Journal of Combinatorial Theory, Series A*, 73(1), pp. 172–173. (1996).
16. F. V. Fomin, D. Lokshtanov, S. Saurabh. Efficient Computation of Representative Sets with Applications in Parameterized and Exact Algorithms. SODA 2014, pp. 142–151. (2014).

17. H. Gao, F. K. Hwang, M. T. Thai, W. Wu, T. Znati. Construction of $d(H)$ -disjunct matrix for group testing in hypergraphs. *J. Comb. Optim.* 12(3), pp 297–301. (2006).
18. A. Gabizon, D. Lokshtanov, M. Pilipczuk. Fast Algorithms for Parameterized Problems with Relaxed Disjointness Constraints. (ESA 15) *CoRR*, abs/arXiv:1411.6756, 2015
19. P. Indyk, H. Q. Ngo, A. Rudra. Efficiently decodable non-adaptive group testing. In the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 10), pp. 1126–1142. (2010).
20. W. H. Kautz, R. C. Singleton, Nonrandom binary superimposed codes, *IEEE Trans. Inform. Theory*, 10(4), pp. 363–377. (1964).
21. I. Koutis, Faster algebraic algorithms for path and packing problems, in Proc. 35th International Colloquium on Automata, Languages and Programming, ICALP 2008, pp. 575–586.
22. N. Linial, M. Luby, M. E. Saks, D. Zuckerman. Efficient Construction of a Small Hitting Set for Combinatorial Rectangles in High Dimension. *Combinatorica* 17(2): pp. 215–234 (1997)
23. L. Liu, H. Shen. Explicit constructions of separating hash families from algebraic curves over finite fields. *Designs, Codes and Cryptography*, 41(2), pp. 221–233. (2006).
24. A. J. Macula, L. J. Popyack. A group testing method for finding patterns in data. *Discret Appl Math.* 144. pp. 149–157. (2004).
25. A. J. Macula, V. V. Rykov, S. Yekhanin. Trivial two-stage group testing for complexes using almost disjunct matrices. *Discrete Applied Mathematics*. 137(1), pp. 97–107. (2004).
26. H. Q. Ngo, D. Z. Du. A survey on combinatorial group testing algorithms with applications to DNA library screening. *Theoretical Computer Science*, 55, pp. 171–182. (2000).
27. J. Naor, M. Naor. Small-bias probability spaces: efficient constructions and applications. *SIAM J. Comput.*, 22(4), pp. 838–856. (1993).
28. M. Naor, L. J. Schulman, A. Srinivasan. Splitters and Near-optimal Derandomization. FOCS 95, pp. 182–191, (1995).
29. E. Porat, A. Rothschild. Explicit Nonadaptive Combinatorial Group Testing Schemes. *IEEE Transactions on Information Theory* 57(12), pp. 7982–7989 (2011).
30. D. R. Stinson, T. Van Trung, R. Wei. Secure Frameproof Codes, Key Distribution Patterns, Group Testing Algorithms and Related Structures, *Journal of Statistical Planning and Inference*, 86, pp. 595–617, (1997).
31. D. R. Stinson, R. Wei, L. Zhu. New constructions for perfect hash families and related structures using combinatorial designs and codes, *J. Combin. Designs.*, 8(3), pp. 189–200. (2000).
32. D. R. Stinson, R. Wei, L. Zhu. Some new bounds for cover-free families, *Journal of Combinatorial Theory, Series A*, 90(1), pp. 224–234. (2000).
33. D. C. Torney. Sets pooling designs. *Ann. Comb.* 3, pp. 95–101.(1999).

A Technical results for the proof of Lemma 4

In this appendix we give some proofs of technical results needed for Lemma 4.

Here we assume that r and k are large enough integers

Lemma 11. *Let $z = 16r \log k / (k \log(4r/k))$. Then z is a monotonically decreasing function in k in the interval $[\sqrt{r}, r]$. In particular, $16\sqrt{r} \geq z \geq 8 \log r$.*

Proof. From $\partial z / \partial k|_{k=x} = 0$ we get $\ln^2 x - (\ln 4r) \ln x + \ln(4r) = 0$. This gives two solutions x_0, x_1 for x . One satisfies $\ln x_0 > \ln r$ and therefore $x_0 > r$ and the second $\ln x_1 < 2$ and therefore $x_1 < e^2 < \sqrt{r}$. This implies that the function is monotone in the interval $[\sqrt{r}, r]$. Now since $z|_{k=\sqrt{r}} = 16\sqrt{r}$ and $z|_{k=r} = 8 \log r$ the result follows.

We remind the reader that

$$\sigma(r, k) := \left(\frac{2\pi r}{k} \right)^{k/2} e^{k^2/(12r)}.$$

Lemma 12. *Let $z = 16r \log k / (k \log(4r/k)) > 8$ and $k = \omega(\sqrt{r})$.*

$$\sqrt{\frac{r}{z}} \cdot \sigma\left(\frac{r}{z}, \frac{k}{z}\right) \left(\frac{r}{z}\right)^{4r/z} \log \frac{r}{z} = o(\sigma(r, k)).$$

Proof. First

$$\sigma\left(\frac{r}{z}, \frac{k}{z}\right) = \sigma(r, k)^{1/z} \leq \sigma(r, k)^{\frac{1}{8}}. \quad (8)$$

Now

$$\begin{aligned} \sqrt{\frac{r}{z}} \cdot \left(\frac{r}{z}\right)^{4r/z} \log \frac{r}{z} &\leq \left(\frac{r}{z}\right)^{5r/z} \\ &= \left(\frac{k \log \frac{4r}{k}}{16 \log k}\right)^{5r/z} \\ &\leq k^{5r/z} \\ &= k^{\frac{5k \log \frac{4r}{k}}{16 \log k}} \\ &= \left(\frac{4r}{k}\right)^{\frac{5k}{16}} \leq \sigma(r, k)^{\frac{5}{8}}. \end{aligned}$$

This with (8) implies the result.

Lemma 13. *Let $z = 16r \log k / (k \log(4r/k))$. Then*

$$\binom{r^2}{z-1} \left(c_1 \left(\frac{r}{z}\right)^{2.5} \sigma\left(\frac{r}{z}, \frac{k}{z}\right) \log \frac{r}{z} \log r \right)^z = r^{O(z)} \cdot \sigma(r, k)$$

Proof. First we have

$$\sigma\left(\frac{r}{z}, \frac{k}{z}\right)^z = \sigma(r, k).$$

Now

$$\binom{r^2}{z-1} \left(c_1 \left(\frac{r}{z}\right)^{2.5} \log \frac{r}{z} \log r \right)^z \leq \left(\frac{er^2}{z}\right)^z r^{4.5z} \leq r^{7z}.$$

Lemma 14. *Let $z = 16r \log k / (k \log(4r/k))$. For $r \geq k = \omega(\sqrt{r})$ we have*

$$r^{O(z)} = \sigma(r, k)^{o(1)}$$

Proof. Let $k = \sqrt{r} \cdot \phi(r)$ where $\phi(r) = \omega(1)$. Then for a constant c there is a constant c' such that

$$\log r^{cz} \leq c' \frac{\sqrt{r} \log^2 r}{\phi(r) \log(\sqrt{r}/\phi(r))}$$

and there is a constant c'' such that

$$\log \sigma(r, k) \geq c'' \phi(r) \sqrt{r} \log \frac{\sqrt{r}}{\phi(r)}.$$

Now, for some constant c''' ,

$$\frac{\log r^{cz}}{\log \sigma(r, k)} \leq c''' \frac{\log^2 r}{\phi^2(r) \log^2(\sqrt{r}/\phi(r))} = o(1).$$

B Application to parametrized algorithms with relaxed disjointness constraints

In this appendix, for the purpose of deriving Theorems 2 and 3, we explain how objects related to cover-free families were used by [18] to obtain certain parameterized algorithms.

Notation. Throughout this appendix, we use the notation O_k to hide $k^{O(1)}$ terms. We denote $[n] = \{1, 2, \dots, n\}$. For sets A and B , by $\{A \rightarrow B\}$ we denote the set of all functions from A to B . The notation \triangleq is used to introduce new objects defined by formulas on the right hand side.

In fact, [18] do not use CFFs directly, but related objects called *minimal separating families* (Definition 2) that have an additional injectivity property. We begin by formally showing that CFFs indeed imply minimal separating families of similar size.

B.1 From CFFs to minimal separating families

Hashing families. Recall that, for an integer $t \geq 1$, we say that a family of functions $\mathcal{H} \subseteq \{[n] \rightarrow [m]\}$ is a *t-perfect hash family*, if for every $C \subseteq [n]$ of size $|C| = t$ there is $f \in \mathcal{H}$ that is injective on C . Alon, Yuster and Zwick [5] used a construction of Moni Naor (based on ideas from Naor et al. [28]) to hash a subset of size t into a world of size t^2 using a very small set of functions:

Theorem 4 ([5] based on Naor). *For integers $1 \leq t \leq n$, a t-perfect hash family $\mathcal{H} \subseteq \{[n] \rightarrow [t^2]\}$ of size $t^{O(1)} \cdot \log n$ can be constructed in time $O(t^{O(1)} \cdot n \cdot \log n)$*

We will also use the following perfect hash family given by Naor, Schulman and Srinivasan [28].

Theorem 5 ([28]). *For integers $1 \leq t \leq n$, a t -perfect hash family $\mathcal{H} \subseteq \{[k^2] \rightarrow [t]\}$ of size $e^{t+O(\log^2 t)} \cdot \log k$ can be constructed in time $O(e^{t+O(\log^2 t)} \cdot k \cdot \log k)$.*

Definition 2 (Minimal separating family). *A family of functions $\mathcal{H} \subseteq \{[n] \rightarrow [t+1]\}$ is (t, k) -minimal separating if for every disjoint subsets $C, D \subseteq [n]$ with $|C| = t$ and $|D| \leq k - t$, there is a function $h \in \mathcal{H}$ such that*

- $h(C) = [t]$.
- $h(D) \subseteq \{t+1\}$.

We show that small cover-free families imply small minimal-separating families.

Lemma 15. *Fix any $t \leq k \leq n$. Suppose a $(k^2, (t, k-t))$ -CFF \mathcal{F} can be constructed in time S . Then a (t, k) -minimal separating family of size $O_k(|\mathcal{F}| \cdot 2^{O(t)})$ can be constructed in time $O_k(S \cdot 2^{O(t)} \cdot \log n \cdot n)$.*

Proof. Fix disjoint subsets $C, D \subseteq [n]$ with $|C| = t$ and $|D| \leq k - t$. It will be convenient to present the family by constructing h adaptively given C and D . That is, for arbitrarily chosen C and D , we will adaptively construct a function h that separates C from D . Function h will be constructed by taking a number of *choices*, where each choice is taken among a number of possibilities. The final family \mathcal{H} will comprise all h that can be obtained using any such sequence of choices; thus, the product of the numbers of possibilities will limit the size of \mathcal{H} . As C and D are taken arbitrarily, it immediately follows that such \mathcal{H} separates every pair (C, D) .

1. Let $\mathcal{H}_0 \subseteq \{[n] \rightarrow [k^2]\}$ be the k -perfect hash family given by Theorem 4. Choose $f_0 \in \mathcal{H}_0$ that is injective on $C \cup D$ — there are $k^{O(1)} \cdot \log n$ choices for this stage.
From now on, we identify C and D with their images in $[k^2]$ under f_0 .
2. Note that an element $f \in \mathcal{F}$ can be viewed as a function $f : [k^2] \rightarrow \{0, 1\}$. Now choose an element f_1 of the $(k^2, (t, k-t))$ -CFF \mathcal{F} , with $f_1(C) \equiv 1$ and $f_1(D) \equiv 0$ — there are $|\mathcal{F}|$ choices for this stage.
At this stage we have ‘separated’ C from D , and just need to satisfy the additional requirement of being injective on C .
3. Let $\mathcal{H}_2 \subseteq \{f_1^{-1}(1) \rightarrow [t]\}$ be the t -perfect hash family given by Theorem 5. Choose a function $f_2 \in \mathcal{H}_2$ that is injective on C — there are $e^{t+O(\log^2 t)} \cdot \log k$ choices for this stage.

The running times and family size are immediate from the construction.

Plugging in our construction from Theorem 1 to the above we get

Corollary 1. *Fix any $t \leq k \leq n$. A (t, k) -minimal separating family of size $O_k((k/t)^{t+o(t)} \cdot 2^{O(t)} \cdot \log n)$ can be constructed in time $O_k((k/t)^{t+o(t)} \cdot 2^{O(t)} \cdot \log n \cdot n)$.*

Proof. It's a straightforward plugin of Theorem 1 into Lemma 15. The only thing to notice is that for any $t \leq k$,

$$N(t, k - t)^{1+o(1)} \cdot 2^{O(t)} \leq (k \cdot (ek/t)^t)^{1+o(1)} = O_k((k/t)^{t+o(t)} \cdot 2^{O(t)}).$$

We proceed to define and construct *multiset separators* that are smaller than those in [18].

B.2 Multiset Separators

Notation for multisets. Fix integers $n, r, k \geq 1$. We use $[r]_0$ to denote $\{0, \dots, r\}$. An r -set is a multiset A where each element of $[n]$ appears at most r times. It will be convenient to think of A as a vector in $[r]_0^n$, where A_i denotes the number of times i appears in A . We denote by $|A|$ the number of elements in A counting repetitions. That is, $|A| = \sum_{i=1}^n A_i$. We refer to $|A|$ as the *size* of A . An (r, k) -set is an r -set $A \in [r]_0^n$, where the number of elements with repetitions is at most k . That is, $|A| \leq k$. For two multisets A, B over $[n]$,

Fix r -sets $A, B \in [r]_0^n$. We say that $A \leq B$ when $A_i \leq B_i$ for all $i \in [n]$. By $\overline{A} \in [r]_0^n$ we denote the “complement” of r -set A , that is, $\overline{A}_i = r - A_i$ for all $i \in [n]$. By $A + B$ we denote the “union” of A and B , that is, $(A + B)_i = A_i + B_i$ for all $i \in [n]$. Suppose now that A and B are (r, k) -sets. We say that A and B are (r, k) -compatible if $A + B$ is also an (r, k) -set, and $|A + B| = k$. That is, the total number of elements with repetitions in A and B together is k and any specific element $i \in [n]$ appears in A and B together at most r times. With the notation above at hand, we can define the central object needed for the algorithms of [18].

Definition 3 (Multiset separator). Let \mathcal{F} be a family of r -sets. We say that \mathcal{F} is an (r, k) -separator if for any (r, k) -sets $A, B \in [r]_0^n$ that are (r, k) -compatible, there exists $F \in \mathcal{F}$ such that $A \leq F \leq \overline{B}$.

[18] showed that a minimal separating family can be used to construct an (r, k) -separator.

Theorem 6. [[18] Theorem 3.3] Fix integers n, r, k such that $1 < r \leq k \leq n$, and let $t \triangleq \lfloor 2k/r \rfloor$. Suppose a (t, k) -minimal separating family $\mathcal{H} \subseteq \{[n] \rightarrow [t+1]\}$ can be constructed in time $f(r, k, n)$. Then an (r, k) -separator \mathcal{F} of size $|\mathcal{H}| \cdot (r+1)^t$ can be constructed in time $O_k(f(r, k, t)) \cdot (r+1)^t$.

Plugging in our construction of minimal separating families from Corollary 1 we get

Corollary 2. Fix integers n, r, k such that $1 < r \leq k$. Then an (r, k) -separator \mathcal{F} of size $O_k(r^{4k/r+o(k/r)} \cdot 2^{O(k/r)} \cdot \log n)$ can be constructed in time $O_k(r^{4k/r+o(k/r)} \cdot 2^{O(k/r)} \cdot n \cdot \log n)$.

The above corollary is an analog of Corollary 3.4 in [18] where the exponent of r was $6k/r$ rather than $4k/r + o(k/r)$.

From this point on we do not give full details, as our theorems follow by a direct plug in of Corollary 2 in [18] as a replacement for their Corollary 3.4.

Specifically, using Corollary 2, the algorithm in Corollary 3.8 of [18] for finding a representative set of a family of multisets \mathcal{P} will run in time $O_k(|\mathcal{P}| \cdot r^{4k/r+o(r)} \cdot 2^{O(k/r)} \cdot n \log n)$ rather than $O_k(|\mathcal{P}| \cdot r^{6k/r} \cdot 2^{O(k/r)} \cdot n \log n)$ which will translate to the running times stated in Theorems 2 and 3 when running the Algorithms proving Theorems 5.6 and 5.8 in [18].